

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-188984
(P2001-188984A)

(43) 公開日 平成13年7月10日 (2001.7.10)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 8 G	1/00	C 0 8 G 1/00	D 3 F 0 2 2
B 6 5 G	1/137	B 6 5 G 1/137	A 5 B 0 4 9
G 0 6 F	17/00	G 0 6 F 15/20	Z 5 B 0 5 6
	17/10	15/31	Z 5 H 1 8 0

審査請求 未請求 請求項の数 9 O L (全 10 頁)

(21) 出願番号 特願平11-372208

(22) 出願日 平成11年12月28日 (1999. 12. 28)

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式
社

神奈川県横浜市中区尾上町6丁目81番地

(72) 発明者 横川 宗治

神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式
社内

(74) 代理人 100083552

弁理士 秋田 収喜

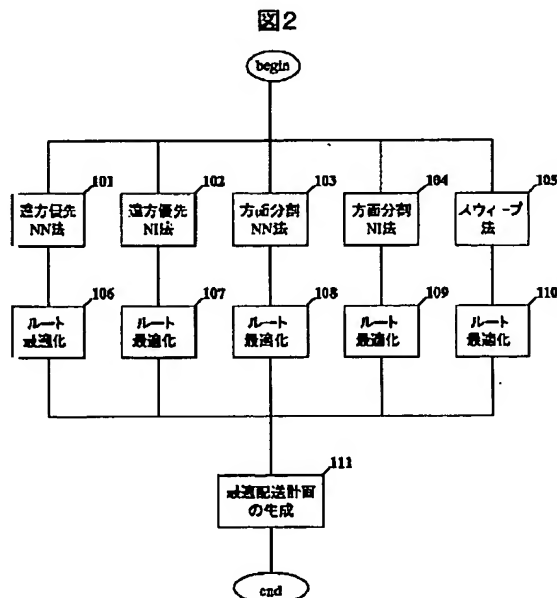
最終頁に続く

(54) 【発明の名称】 最適配送計画立案方法およびシステム

(57) 【要約】

【課題】 あらゆる配送先の位置の分布に対して有効な配送計画を立案可能にすること。

【解決手段】 配送先の位置情報等の配送計画立案に関わる入力情報を入力する入力装置と、入力された情報によって複数の配送車両が物流拠点を出発し、複数の配送先に積み荷を配送した後、元の物流拠点に戻る経路で総走行距離が最短になるようにする配送計画を作成する処理装置と、作成された配送計画を出力する出力装置とを備えた配送計画立案システムにおいて、立案戦略が異なるアルゴリズムの複数の配送計画立案処理を単一の処理装置内で時分割または複数の処理装置で並行に実行させ、その実行結果のうち総走行距離が最短になる処理結果を選択し、前記出力装置から出力する。



【特許請求の範囲】

【請求項1】 配送先の位置情報、荷物量、配送車両数等の配送計画立案に関わる入力情報を入力する入力装置と、入力された情報によって複数の配送車両が物流拠点を出発し、複数の配送先に積み荷を配送した後、元の物流拠点に戻る経路で総走行距離が最短になるようにする配送計画を作成する処理装置と、作成された配送計画を出力する出力装置とを備えた配送計画立案システムにおける最適配送計画立案方法であって、立案戦略が異なるアルゴリズムの複数の配送計画立案処理を単一の処理装置内で時分割または複数の処理装置で並行に実行させ、その実行結果のうち総走行距離が最短になる処理結果を選択し、前記出力装置から出力することを特徴とする最適配送計画立案方法。

【請求項2】 前記複数の配送計画立案処理の1つとして、配送車両が最初に巡る配送先として未配送の配送先の中で物流拠点から最も遠い配送先を選択し、それ以後、当該配送先に最も近い配送先を順次選択して配送経路を探索するアルゴリズムで構成された配送計画立案処理を用いることを特徴とする請求項1記載の最適配送計画立案方法。

【請求項3】 前記複数の配送計画立案処理の1つとして、配送車両が最初に巡る配送先として未配送の配送先の中で物流拠点から最も遠い配送先を選択し、それ以後、配送経路に挿入した時の走行距離の増分が最小である配送先を順次選択して配送経路を探索するアルゴリズムで構成された配送計画立案処理を用いることを特徴とする請求項1記載の最適配送計画立案方法。

【請求項4】 前記複数の配送計画立案処理の1つとして、配送拠点を中心とする全方位の配送対象地域を複数の方面に分割し、各方面ごとに、配送車両が最初に巡る配送先として未配送の配送先の中からその方面内に位置する配送先の荷物量の和に比例する数の配送先を選択し、それ以後、当該配送先に最も近い配送先を順次選択して配送経路を探索するアルゴリズムで構成された配送計画立案処理を用いることを特徴とする請求項1記載の最適配送計画立案方法。

【請求項5】 前記複数の配送計画立案処理の1つとして、配送拠点を中心とする全方位の配送対象地域を複数の方面に分割し、各方面ごとに、配送車両が最初に巡る配送先として未配送の配送先の中からその方面内に位置する配送先の荷物量の和に比例する数の配送先を選択し、それ以後、配送経路に挿入した時の走行距離の増分が最小である配送先を順次選択して配送経路を探索するアルゴリズムで構成された配送計画立案処理を用いることを特徴とする請求項1記載の最適配送計画立案方法。

【請求項6】 前記複数の配送計画立案処理の1つとして、配送先と物流拠点を結ぶ直線と配送区域地図上のx軸またはy軸とのなす角度の小さい順に配送先を選択して配送経路を探索するアルゴリズムで構成された配送計

画立案処理を用いることを特徴とする請求項1記載の最適配送計画立案方法。

【請求項7】 複数の配送計画立案処理の結果に対し、1台の配送車両が物流拠点を出発し、複数の配送先に積み荷を配送した後、元の物流拠点に戻る配送経路上の配送先数が閾値より少ない配送経路に対しては全探索法を、多い配送経路に対しては2OPT法をそれぞれ適用して最適化することを特徴とする請求項1～6のいずれか一項に記載の配送計画立案方法。

【請求項8】 配送先の位置情報、荷物量、配送車両数等の配送計画立案に関わる入力情報を入力する入力装置と、入力された情報によって複数の配送車両が物流拠点を出発し、複数の配送先に積み荷を配送した後、元の物流拠点に戻る経路で総走行距離が最短になるようにする配送計画を作成する処理装置と、作成された配送計画を出力する出力装置とを備えた配送計画立案システムであって、

前記処理装置が、立案戦略が異なるアルゴリズムの複数の配送計画立案処理を時分割で実行し、その実行結果のうち総走行距離が最短になる処理結果を選択し、前記出力装置から出力する処理手段を備えることを特徴とする最適配送計画立案システム。

【請求項9】 配送先の位置情報、荷物量、配送車両数等の配送計画立案に関わる入力情報を入力する入力装置と、入力された情報によって複数の配送車両が物流拠点を出発し、複数の配送先に積み荷を配送した後、元の物流拠点に戻る経路で総走行距離が最短になるようにする配送計画を作成する処理装置と、作成された配送計画を出力する出力装置とを備えた配送計画立案システムであって、

前記処理装置が複数の処理装置で構成され、各処理装置が、立案戦略が異なるアルゴリズムの複数の配送計画立案処理を並列に実行する処理手段を備え、いずれか1つの処理装置が、複数の処理装置の実行結果のうち総走行距離が最短になる処理結果を選択し、前記出力装置から出力する手段を備えることを特徴とする最適配送計画立案システム。

【発明の詳細な説明】

【0001】

【発明の属する分野】本発明は、物流システムにおける物流拠点または倉庫を出発し、複数の配送先に積み荷を配送する配送計画を作成する配送計画立案方法に係わり、特に宅配便などの効率的な配送のための輸送ルートの見込に利用し、物流コストの削減やサービスの向上を実現する配送計画を作成する方法に関する。

【0002】

【従来の技術】物流システムにおける配送計画の問題は、複数の車両が共通の物流拠点を出発し、複数の配送先に積み荷を配送して、元の物流拠点に戻る配送形態で、総走行距離が最短になるようなものを求めるもので

ある。ただし、各配送先には1台の車両だけが通るものとする。また、各車両には積載量の上限があり、その車両が訪れる配送先への荷物量の合計よりも大きいという積載量制約がある。また、各車両の走行距離に上限があるという走行距離制約がある。

【0003】このような制約のもとで作成する配送計画の作成ステップは、通常、2段階に分かれる。1つは配送先を各車両に振り分けるグルーピングの問題である。他の1つはグルーピングで決められた配送先グループを車両がどの順序で巡るかというルート最適化の問題である。これはいわゆる巡回セールスマンの問題とも呼ばれている。

【0004】グルーピングの問題については、スウィープ法やセービング法などが知られている。これらの方法は下記の文献に記述されている。

Gilbert Laporte: The Vehicle Routing Problem: An overview of exact and approximate algorithms: European Journal of Operational Research 59 (1992) 345-358

また、巡回セールスマン問題の解法である全探索法と2OPT法やグルーピングで使われるNN法(Nearest Neighbor法)とNI法(Nearest Insertion法)についての知見は下記の文献で得られる。山本芳嗣、久保幹雄: 巡回セールスマン問題への招待: 朝倉書店(1997)

【0005】

【発明が解決しようとする課題】しかしながら、従来における配送計画方法にあっては、グルーピングを行うために単一のアルゴリズムで対処していた。例えば、グルーピングにNN法を適用すれば、多くの場合、比較的良好な結果を得ることができた。しかし、部品調達のアプリケーションなどでは、一部の配送先が物流拠点から遠距離の場所に位置していることがある。このような場合、グルーピングの問題に単純なNN法を用いると、まず、物流拠点から比較的近距離にある配送先がグループにまとめられ、その後、物流拠点から遠距離にある配送先同志がグループにまとめられる傾向がある。このため、この方法では全体として総走行距離が長くなってしまいう問題がある。一般に、従来の配送計画作成方法は、配送先の位置の分布によってグルーピングの方法に向き不向きがあるため、不適当な結果となることがある。

【0006】また、ルートの最適化についても1つのアルゴリズムが用いられていた。しかし、1つのルートに含まれる配送先の数は様々であり、例えば、2OPT法は配送先の数が多い場合には効率が良いが、数が少ない場合には他の方法と比較して精度が落ちるという問題がある。

【0007】本発明の目的は、あらゆる配送先の位置の分布に対して有効な配送計画を立案することができる最適配送計画立案方法およびシステムを提供することであ

る。

【0008】

【課題を解決するための手段】上記目的を達成するために、本発明の最適配送計画立案方法は、複数のグルーピング戦略を用いて複数の配送計画を生成し、その中で最適な配送計画を選択するようにしたことを特徴とする。

【0009】すなわち、配送先の位置情報、荷物量、配送車両数等の配送計画立案に関わる入力情報を入力する入力装置と、入力された情報によって複数の配送車両が物流拠点を出発し、複数の配送先に積み荷を配送した後、元の物流拠点に戻る経路で総走行距離が最短になるようにする配送計画を作成する処理装置と、作成された配送計画を出力する出力装置とを備えた配送計画立案システムにおける最適配送計画立案方法であって、立案戦略が異なるアルゴリズムの複数の配送計画立案処理を単一の処理装置内で時分割または複数の処理装置で並行に実行させ、その実行結果のうち総走行距離が最短になる処理結果を選択し、前記出力装置から出力することを特徴とする。

【0010】本発明では、例えば次の5つのグルーピング戦略に基づいたアルゴリズムを使用する。

- (1) 遠方優先NN法
- (2) 遠方優先NI法
- (3) 方面分割NN法
- (4) 方面分割NI法
- (5) スウィープ法

【0011】ここで、「遠方優先」と「方面分割」という言葉は配送車両が最初に巡る配送先、すなわちシードの指定方法を表す。また、NN法、NI法とスウィープ法はいくつかの配送先を巡ってルートを生成するアルゴリズムである。したがって、「遠方優先NN法」とは、シードの指定の仕方が「遠方優先」で、配送先を巡るアルゴリズムが「NN」法であるグルーピング方法のことである。遠方優先NI法、方面分割NN法、方面分割NI法も同様の意味を持つ。

【0012】遠方優先では、未配送の配送先の中で物流拠点から最も遠い配送先をシードとして指定する。方面分割では、空間(配送対象地域)をいくつかの物流拠点を通る直線で分割して生成される例えば扇形の部分空間(以後、方面と呼ぶ)ごとに、その方面内に位置する配送先の荷物量の和に比例する数のシードをその方面に割り当てる。

【0013】NN法では、シードから始めて、制約条件を満たす間、次々に最後に訪れた配送先に最も近い配送先を訪れる。制約条件が満たされなくなったならば、それまで訪れた配送先で1つのルートの生成を完了する。未配送の配送先が残っている間、前と同様に、シードから始めてルートを生成し続ける。

【0014】NI法では、ルートの適当な位置に挿入した時の走行距離の増分が最小である配送先を取り出す他

はNN法と同様である。

【0015】スウィープ法では、配送先と物流拠点を結ぶ直線とx軸（またはy軸）のなす角度の小さい順に配送先を取り出す他はNN法と同様である。

【0016】また、本発明では、ルートに含まれる配送先の数に応じてルートの最適化の方法を変えるようにしたことを特徴とするものである。具体的には、配送先の数が比較的少ないルートに対しては精度の高い全探索法を、多いルートに対しては高速の2OPT法をそれぞれ適用して、効率的なルートの最適化を実現する。

【0017】

【発明の実施の形態】以下、本発明を図示する実施の形態に基づいて詳細に説明する。図1は本発明の最適配送計画立案システムの一実施形態を示すハードウェア構成のブロック図である。この実施形態の最適配送計画立案システムはキーボード10、ディスプレイ20、中央処理装置30、主記憶装置40、2次記憶装置50から構成されている。中央処理装置30はキーボード10とディスプレイ20が接続されており、キーボード10を用いて配送対象地域における配送先の位置、荷物量、車両の積載上限値、走行距離上限値などのパラメータ情報、ならびに配送計画の作成に関わる各種のコマンドが入力される。入力されたパラメータ情報やコマンドはディスプレイ20に表示される。ディスプレイ20には、入力されたパラメータ情報に基づいて作成された最適配送経路が文字や地図によって表示される。主記憶装置40には、複数の立案戦略によって配送経路を立案するための配送計画立案プログラム群41が記憶されている。さらに、データを処理するための作業領域42が確保されている。2次記憶装置50には、配送先の位置、荷物量や車両の積載量の上限などのデータやコマンドの処理結果が格納される。

【0018】図2は、5つのグルーピング戦略を用いた最適配送計画立案処理の全体構成図である。本実施形態では、最初に、遠方優先NN法の配送計画立案処理（ステップ101）、遠方優先NI法の配送計画立案処理（ステップ102）、方面分割NN法の配送計画立案処理（ステップ103）、方面分割NI法の配送計画立案処理（ステップ104）、スウィープ法の配送計画立案処理（ステップ105）によって、配送先を各車両に振り分ける。次に、ルート最適化処理（ステップ106～110）で、配送先グループごとに車両のルートが最短になるようにする配送計画を作成する。最後に、この5つの配送計画の中で最適な配送計画を選択し（ステップ111）、処理を終える。

【0019】ここで、ステップ101～105の5つの配送計画立案処理は、対応するプログラムによって構成され、図1の配送計画立案プログラム群41として主記憶装置40に格納されている。この実施形態にあっては、処理装置30が1台であるので、中央処理装置30

が時分割で5つの配送計画立案処理を行い、さらにルートの最適化処理（ステップ106～110）を時分割で行い、最後に、5つの配送計画立案処理結果の中で最適な配送計画を選択する。

【0020】本発明は、5つの配送計画立案処理を時分割で行うものに限定されるものではなく、中央処理装置30を5台用意し、これらによって並列に立案処理及び最適化処理を実行させ、1つの中央処理装置が5つの配送計画立案結果の中で最適な配送計画を選択するように構成することもできる。

【0021】図3は遠方優先NN法の配送計画立案処理のアルゴリズムを説明するフローチャートである。まず、ステップ201で車両tを0番目の車両に設定する。次にステップ202で、未配送の配送先があるか判定する。なければ、終了する。あれば、ステップ203で、未配送の配送先で最も物流拠点から遠いものを配送先iとし、この配送先iを配送済にする。配送先iは遠方優先のシードである。個々の車両tについての情報は構造体の配列T[t]の要素で表現する。

- (1) T[t].Lは車両tの運ぶ荷物量である。
- (2) T[t].Dは車両tの走行距離である。
- (3) T[t].Rは配送先の配列で、車両tのルートである。

【0022】ステップ204で、車両tの物流拠点とシードを結ぶルートを次のようにして生成する。

- (1) T[t].Lに配送先iの荷物量を代入する。
- (2) T[t].Dに配送先iと物流拠点の間の距離を2倍したものを代入する。
- (3) T[t].Rの先頭に配送先iを代入する。

【0023】次にステップ205で、未配送の配送先があるか判定する。なければ、終了する。あれば、ステップ206で、未配送の配送先で最も配送先iに近くのを配送先iiとする。次にステップ207で、配送先iiを車両tのルートに追加した時の走行距離の増分 $D=d(i, ii)+d(\text{depot}, ii)d(\text{depot}, i)$ を計算する。ここでdは距離関数であり、depotは物流拠点を示す。

【0024】次にステップ208で、積載量制約と走行距離制約を満足するか判定する。積載量制約は車両tが配送先iiの荷物を積み込んでも、与えられた積載量の上限は越えないという制約である。また、走行距離制約は車両tがさらに走行距離をDだけ増やしても、与えられた走行距離の上限を越えないという制約である。この2つの制約を満足すれば、ステップ209で、配送先iiを車両tが通るように設定する。すなわち、構造体の配列T[t]を次のように変更する。

- (1) T[t].Lに配送先iiの荷物量を加える。
- (2) T[t].DにDを加える。
- (3) T[t].Rの最後に配送先iiを追加する。

【0025】さらに、配送先iiを配送済にする。次にス

ステップ210で、配送先 i に配送先 i を代入して、ステップ205に戻り、ループする。ステップ208で、2つの制約を満足しなければ、ステップ211で、車両 t を次の車両にして、ステップ202に戻り、ループする。

【0026】図4は、遠方優先NI法のアルゴリズムを説明するフローチャートである。ステップ301で、車両 t を0番目の車両に設定する。次にステップ302で、遠方優先のシード i を獲得し、シード i を配送済にする。次にステップ303で、車両 t がシード i を通るように設定する。ステップ304以後のアルゴリズムには3重のループがある。ステップ304、317のループと i に関するステップ306、307、316のループと j に関するステップ309、310、315のループである。

【0027】ステップ304で、未配送の配送先があるか判定する。なければ、ループを抜けて終了する。あれば、ステップ305で、変数 \min_d に適当な大きな定数MAXを代入する。次にステップ306で、配送先 i に0番目の配送先を代入する。ステップ307で、 i と全配送先の数 \max_i を比較する。 i が \max_i より小さくなければ、 i に関するループを抜けてステップ317に進む。小さければ、ステップ308で、配送先 i が配送済であるか判定する。配送済であれば、ステップ316に進む。未配送であれば、 j に関するループに入る。まずステップ309で、変数 j に0を代入する。次に、ステップ310で、変数 j と配列 $T[t].R$ の長さを比較する。 j が大きければ、 j に関するループを抜けてステップ316に進む。 j が小さくなければ、ステップ311で、走行距離の増分 D を計算する。前に述べたように、 $T[t].R$ は配送先の配列であって、車両 t が通るルートを表す。 $T[t].R[j]$ は車両 t が物流拠点を出発してから $(j+1)$ 番目に訪れる配送先である。 D は j 番目と $(j+1)$ 番目の間に配送先 i を挿入した時の走行距離の増分を表す。式で表すと

$$D = d(T[t].R[j], i) + d(i, T[t].R[j]) - d(T[t].R[j], T[t].R[j])$$

となる。

【0028】次にステップ312で、変数 \min_d と走行距離の増分 D を比較する。 \min_d が D より大きくなければ、ステップ315に進み、 j に関するループを繰り返す。 \min_d が D より大きければ、ステップ313に進み、積載量制約と走行距離制約を満たしているか判定する。満たさなければ、ステップ315に進み、 j に関するループを繰り返す。満たせば、この時の D 、 i 、 j をそれぞれ \min_d 、 \min_i 、 \min_j に代入して、ステップ315に進み、 j に関するループを繰り返す。結局ここで、 i と j に関する D の最小値を獲得する。

【0029】図5は、ステップ307の i に関するループを抜けて、次に実行するステップ317の詳細を示す

フローチャートである。まず、ステップ401で、 \min_d がMAXであるか判定する。MAXであれば、 \min_d が変わらなかったこと、すなわち、挿入すべき配送先が見つからなかったことを意味するので、車両 t のルートを延長するのを止める。そしてステップ402で、車両 t を次の車両にする。さらにステップ403で、遠方優先のシード i を獲得し、このシード i を配送済にする。次にステップ404で、車両 t がシード i を通るように設定する。一方、 \min_d がMAXでなければ、ステップ405で、車両 t が配送先 \min_i を (\min_j+1) 番目に訪れるように設定し、配送先 \min_i を配送済にする。

【0030】方面分割NN法は方面に車両を割り当てる手続きallocateとreallocateを呼ぶ。以下では、先にallocateとreallocateのアルゴリズムの説明をする。

【0031】図6は、allocateのアルゴリズムを説明するフローチャートである。ステップ501で各配送先が含まれる方面 i を計算し、配送先の荷物量を加えることによって $FL[i]$ を求める。次にステップ502で車両台数の初期値 N を各方面 i に $FL[i]$ に比例した台数 $FT[i]$ を割り当てる。ただし、ここで除算の値は整数である。また今割り当てた台数 $FT[i]$ の i についての和 TFT を求める。次にステップ503で除算の剰余を $W[i]$ に入れる。そして、ステップ504と505でまだ割り当てられていない残りの $(N-TFT)$ 台の車両を $W[i]$ の大きい順に方面 i に1台ずつ割り当てる。すなわち、 $FT[i]$ を1だけ増やす。次にステップ506で各方面 i の車両台数 $FT[i]$ 個の配送先をランダムに選び、これをシードとして物流拠点とシードを結ぶルートを生成する。さらに、配送 j を配送済みにする。最後にステップ507で変数 \max_{truck} に N を代入する。

【0032】reallocateは有効な車両が尽きた時、総残荷物量 TRL に比例した台数の車両を獲得して各方面に割り当てる。reallocateは次の2点を除いて基本的にはallocateと同じである。第1に、全体の割り当て車両台数は初期値 N ではなく、総残荷物量 TRL を車両の積載量の上限で割った数 IN である。第2に、各方面 i の車両台数は各方面の荷物量 $FL[i]$ ではなく、残荷物量 $RFL[i]$ に比例している。なお、総残荷物量 TRL は未配送の配送先の荷物量の総計であり、各方面 i の残荷物量 $RFL[i]$ は方面 i にある未配送の配送先の荷物量の総計である。

【0033】図7は、方面分割NN法のアルゴリズムを説明するフローチャートである。方面分割NN法は2重のループを含む。外側のループは全ての配送先が車両のルートに含まれた時、終了する。内側のループはステップ603で車両 t が0から始めて、全ての配送先が車両のルートに含まれた時、または、車両 t が \max_{truck} になった時、終了する。まず、ステップ601でallocateを呼んで、全体で N 台の車両を方面に割り当てる。次に、ステップ602で未配送の配送先があるか判定する。なければ、終了する。あれば、ステップ603で車両 t を

0に設定する。

【0034】次に、ステップ604で未配送の配送先があり、車両 t が maxtruck より小さいか判定する。そうでなければ、外側のループの先頭ステップ602に戻る。そうであれば、ステップ605で車両 t が有効であるか判定する。初期値では車両 t は有効である。また、ステップ611で車両 t は無効になる。有効でなければ、ステップ615で t の値を1つ増やし、内側のループの先頭ステップ604に戻る。有効ならば、ステップ606で $T[t].R$ の最後を i に、未配送の配送先で最も i に近いものを ii にそれぞれ設定する。次に、ステップ607で走行距離の増分

$$D=d(i, ii)+d(\text{depot}, ii)-d(\text{depot}, i)$$

を計算する。

【0035】次に、ステップ608で積載量制約と走行距離制約を満足するか判定する。この2つの制約を満足すれば、ステップ609で配送先 ii を車両 t が通るように設定する。さらに、配送先 ii を配送済みにする。次に、ステップ610で t の値を1つ増やす。2つの制約を満足しなければ、ステップ611で t を無効にし、ステップ612で他に有効な車両があるか判定する。あれば、ステップ613で t の値を1つ増やす。なければ、ステップ614 reallocate を呼んで、車両を再割り当てる。いずれの場合も、内側のループの先頭ステップ604に戻る。

【0036】図8は、方面分割NI法のアルゴリズムを説明するフローチャートである。まず、ステップ701で前に述べた allocate を呼んで、全体で N 台の車両を方面に割り当てる。次に、ステップ702で未配送の配送先があるか判定する。なければ、終了する。あれば、ステップ703で車両 t を0に設定する。次に、ステップ704で未配送の配送先があり、車両 t が maxtruck より小さいか判定する。そうでなければ、ステップ702に戻る。そうであれば、ステップ705で車両 t が有効であるか判定する。初期値では車両 t は有効である。また、ステップ718で車両 t は無効になることがある。有効でなければ、ステップ719で t の値を1つ増やし、ステップ704に戻る。有効ならば、ステップ706に進む。

【0037】ステップ706～717は遠方優先NI法のステップ305～316とまったく同一である。 D は車両 t が通るルートの j 番目と $(j+1)$ 番目の配送先間に配送先 i を挿入した時の走行距離の増分である。ここでは i と j に関する2重のループによって、 D の最小値を獲得し、その時の D , i , j をそれぞれ min_d , min_i , min_j に代入して、ステップ718に進む。

【0038】図9は、ステップ708の i に関するループを抜けて、次に実行するステップ718の詳細を示すフローチャートである。まず、ステップ801で、 min_d が MAX であるか判定する。 MAX であれば、 min_d が変わら

なかったこと、すなわち、挿入すべき配送先が見つからなかったことを意味するので、車両 t のルートを延長するのを止めるため、ステップ802で車両 t を無効にする。次にステップ803で、他に有効な車両があるか判定する。あれば、ステップ804で t の値を1つ増やす。なければ、ステップ805で前に述べた reallocate を呼んで、車両を再割り当てる。

【0039】一方、 min_d が MAX でなければ、ステップ806で、車両 t が配送先 min_i を (min_j+1) 番目に訪れるように設定し、配送先 min_i を配送済にする。次にステップ807で、 t の値を1つ増やす。

【0040】図10は、スウィープ法のアルゴリズムを説明するフローチャートである。スウィープ法では車両が角度の小さい順に配送先を通る。配送先の角度とは、配送先と物流拠点を結ぶ直線と x 軸（または y 軸）のなす角度のことである。また x 軸とは、物流拠点を通り、東西方向に向いた直線のことである。

【0041】ステップ901で、車両 t を0番目の車両に設定する。次にステップ902で、未配送の配送先があるか判定する。なければ、終了する。あれば、ステップ903で、未配送の配送先で最も角度の小さいものを配送先 i とし、この配送先 i を配送済にする。ステップ904で、車両 t の物流拠点と配送先 i を結ぶルートを生成する。次にステップ905で、未配送の配送先があるか判定する。なければ、終了する。あれば、ステップ906で、未配送の配送先で最も小さいものを配送先 ii とする。

【0042】次にステップ907で、配送先 ii を車両 t のルートに追加した時の走行距離の増分

$$D=d(i, ii)+d(\text{depot}, ii)-d(\text{depot}, i)$$

を計算する。次にステップ908で、積載量制約と走行距離制約を満足するか判定する。この2つの制約を満足すれば、ステップ909で、配送先 ii を車両 t が通るように設定する。さらに、配送先 ii を配送済にする。次にステップ910で、配送先 i に配送先 ii を代入して、ステップ905に戻り、ループする。ステップ908で、2つの制約を満足しなければ、ステップ911で、車両 t を次の車両にして、ステップ902に戻り、ループする。

【0043】次にルートの最適化について説明する。グルーピングした結果の車両のルートは必ずしも最適なものではない。そこで、ルートに含まれる配送先の順序を並び換え、より短いルートを生成する必要がある。配送先の比較的小さいルートに対しては、全解探索法が完全に最適化された答えを出す。多いルートに対しては、全解探索法では処理が遅すぎる。これに対して2OPT法が精度は落ちるが、処理は高速である。したがって、ルートの最適化は、ルートに含まれる配送先の数がある閾値以下の場合には全解探索法を、閾値を越える場合には2OPT法を適用して最適解を求める。

【0044】この全探索法と2OPT法のアルゴリズムについては、文献によって知られているので、その説明は省略する。

【0045】なお、5つの配送計画立案処理のためのプログラムは、1つの処理装置で時分割で実行させる形態だけに限定されるものではなく、例えば5台の中央処理装置で並列に実行させる形態、2台の中央処理装置で3つと2つの立案処理を分担して実行させる形態であってもよい。

【0046】

【発明の効果】以上の説明から明らかなように、本発明においては、複数の配送計画立案戦略を用いたグルーピングと2種の方法を用いたルート最適化処理によって、配送計画を立案するようにしたため、広範囲の配送先の位置の分布に対して有効な配送計画を立案することができる。

【図面の簡単な説明】

【図1】本発明の最適配送計画立案システムの一実施形態を示すハードウェア構成図である。

【図2】5つのグルーピング戦略を用いた最適配送計画処理の全体構成図である。

【図3】遠方優先NN法のアルゴリズムを示すフローチャートである。

【図4】遠方優先NI法のアルゴリズムを示すフローチャートである。

【図5】図4のステップ317の詳細を示すフローチャートである。

【図6】車両を方面に割り当てるアルゴリズムを示すフローチャートである。

【図7】方面分割NN法のアルゴリズムを示すフローチャートである。

【図8】方面分割NI法のアルゴリズムを示すフローチャートである。

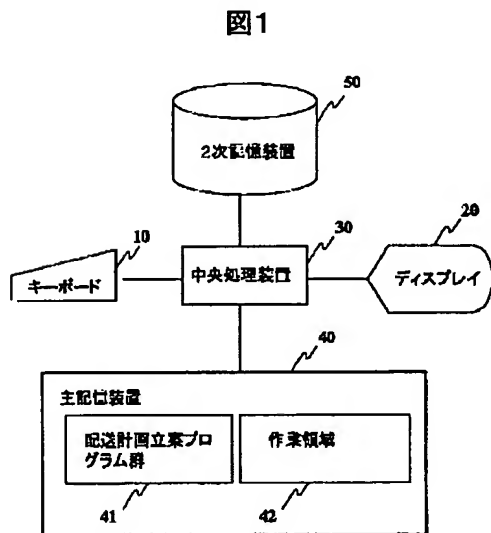
【図9】図8のステップ718の詳細を示すフローチャートである。

【図10】スウィープ法のアルゴリズムを示すフローチャートである。

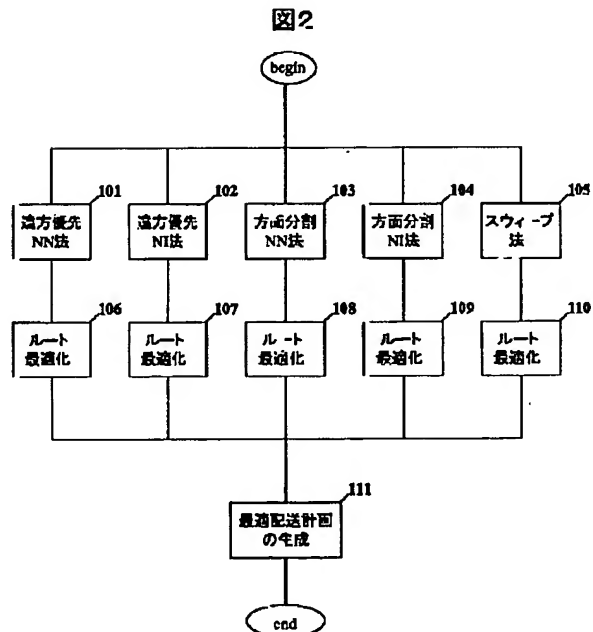
【符号の説明】

10…キーボード、20…ディスプレイ、30…中央処理装置、40…主記憶装置、41…配送計画立案プログラム群、42…作業領域、50…2次記憶装置。

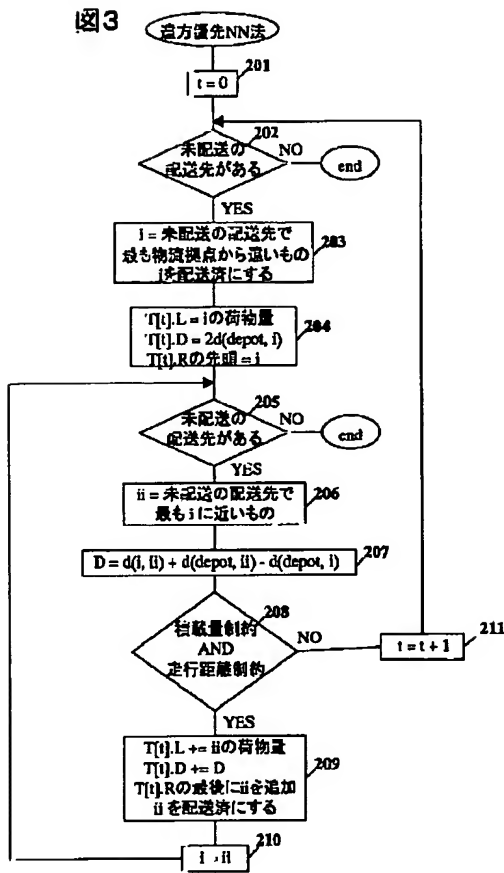
【図1】



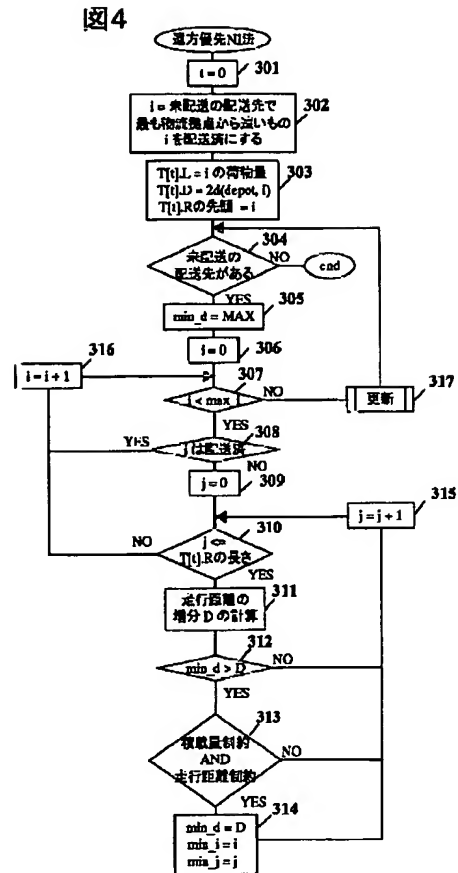
【図2】



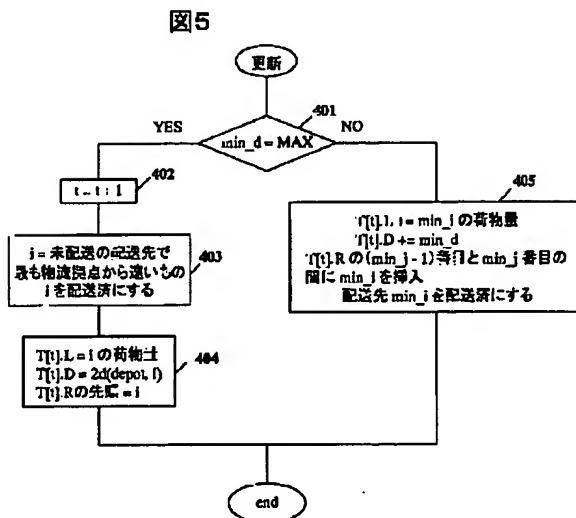
【図3】



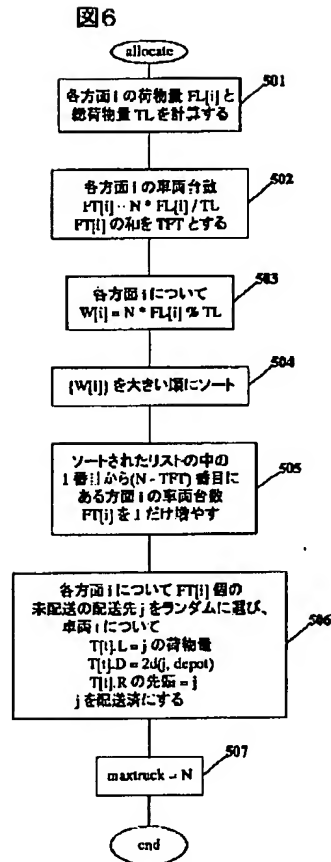
【図4】



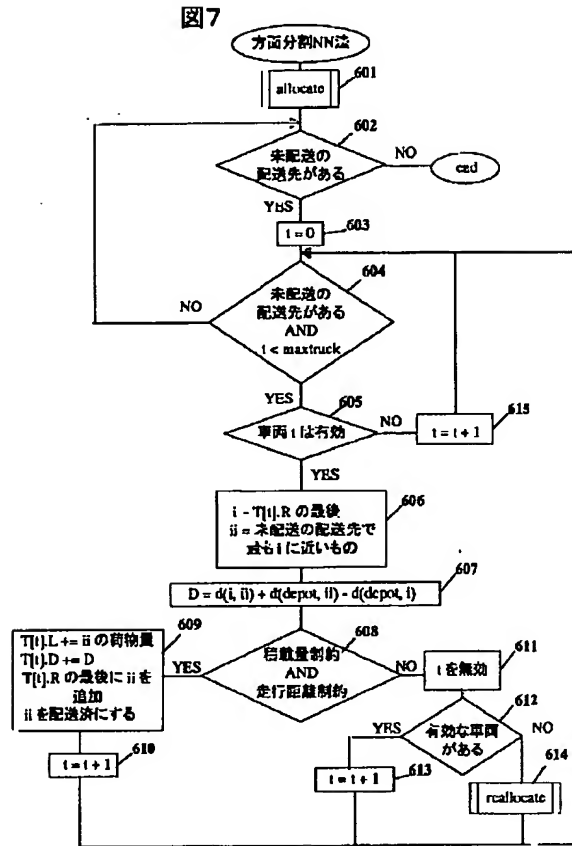
【図5】



【図6】

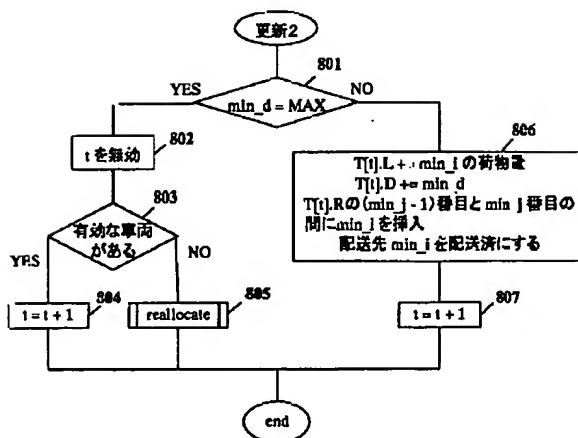


【図7】

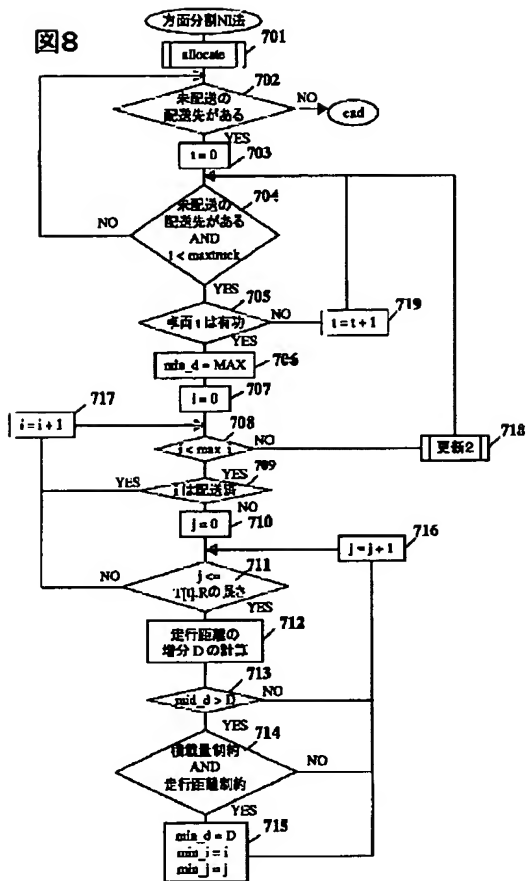


【図9】

図9

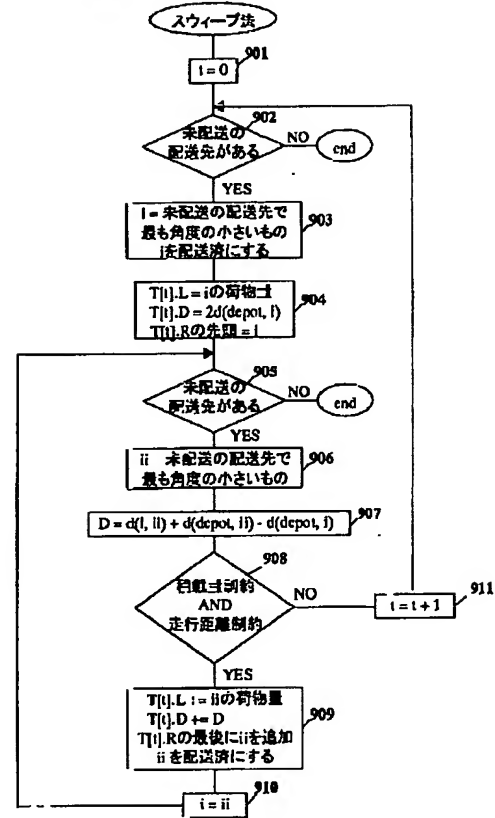


【図8】



【図10】

図10



フロントページの続き

(72)発明者 小野山 隆
神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会
社内

Fターム(参考) 3F022 LL05 MM42 PP03 PP06
5B049 AA02 BB33 CC40 EE31
5B056 AA00 AA04 BB00 HH00
5H180 AA15 EE02 FF01